Statistical Relational Learning for Link Prediction

Alexandrin Popescul and **Lyle H. Ungar**
Department of Computer and Information Sciences
University of Pennsylvania
Philadelphia, PA 19104
{popescul,ungar}@cis.upenn.edu

Abstract

Link prediction is a complex, inherently relational, task. Be it in the domain of scientific citations, social networks or hypertext links, the underlying data are extremely noisy and the characteristics useful for prediction are not readily available in a "flat" file format, but rather involve complex relationships among objects. In this paper, we propose the application of our methodology for *Statistical Relational Learning* to building link prediction models. We propose an integrated approach to building regression models from data stored in relational databases in which potential predictors are generated by structured search of the space of queries to the database, and then tested for inclusion in a logistic regression. We present experimental results for the task of predicting citations made in scientific literature using relational data taken from CiteSeer. This data includes the citation graph, authorship and publication venues of papers, as well as their word content.

1 Introduction

Link prediction is an important problem arising in many domains. Web pages, computers, scientific publications, organizations, people and biological molecules are interconnected and interact in one way or another. Being able to predict the presence of links or connections between entities in a domain is both important and difficult to do well. We emphasize two important characteristics of such domains: i) their nature is inherently multi-relational, making the standard "flat" file domain representation inadequate, and ii) such data is often very noisy or partially observed. For example, in the domain of scientific publications, documents are cited based on many criteria, including their topic, conference or journal, and authorship, as well as the already existing citation structure. All attributes contribute, some in fairly complex ways.

The characteristics of the task suggest statistical learning for building robust models from noisy data and relational databases as a natural way to represent and store such data. Difficulties arise from the fact that the standard statistical learning algorithms assume one-table "flat" domain representation. Such algorithms are presented with a set of possible predictors, and a model selection process makes decisions regarding their inclusion into a model. Thus, the process of feature creation is decoupled from feature selection, and is often performed manually. Moreover, it is not always obvious what features should be generated. Thus, it is crucial to provide statistical modeling techniques with an integrated functionality to navigate richer data structures in order to discover potentially new and complex sources of relevant evidence, features not readily available in single tables and not always immediately obvious to humans.

One approach that might be considered is to generate the full join of a database for a one-table learning method. This is both impractical and incorrect—the size of the resulting table is prohibitive, and the notion of an object corresponding to a training example is lost, being represented by multiple rows. Moreover, the entries in the full join table will be *atomic* attribute values, rather than values resulting from arbitrarily complex queries, what we desire for our features. Preserving the relational representation also allows the introduction of intelligent search heuristics that explore only subspaces of the possible search space.

Our method of statistical relational learning integrates standard statistical modeling, here logistic regression, with a process for systematically generating features from relational data. We formulate the feature generation process as *search in the space of relational database queries*. The richness of this space or *space bias* potentially can be chosen at the desired level of complexity by specifying the types of queries allowed in the search. Aggregation or statistical operations, groupings, richer join conditions, or *argmax*-based queries can all be considered as part of search. Thus, the search space allows the discovery of complex and interesting relationships.

In this paper, we apply our method of statistical relational learning [Popescul *et al.*, 2003] to the problem of citation prediction in the domain of scientific publications. Link prediction models in this domain can be used as a citation recommender service. This service can potentially be deployed to recommend citations to users who provide the abstract, names of the authors and possibly a partial reference list of a paper in progress. In addition to prediction, the learned features have an explanatory power, providing insights into the nature of the citation graph structure. We use the data from CiteSeer (a.k.a. ResearchIndex),[1] an online digital library of computer science papers [Lawrence *et al.*, 1999]. CiteSeer contains a rich set of relational data, including the text of titles, abstracts and documents, citation information, author names and affiliations, conference or journal names.[2]

2 Methodology

Our method of statistical relational learning couples two main processes: generation of feature candidates from relational data and their selection with statistical model selection criteria. Relational feature generation is a search problem. It requires formulation of the search in the space of queries to a relational database. We introduce notation first.

Throughout this section we use the following fixed schema:[3]

Citation(*from*:`Document`, *to*:`Document`),
Author(*doc*:`Document`, *auth*:`Person`),
PublishedIn(*doc*:`Document`, *vn*:`Venue`),
WordCount(*doc*:`Document`, *word*:`Word`, *cnt*:`Int`).

We use extended relational algebra notation to denote aggregations. Aggregation functions are subscripted with the corresponding attribute name if applied to an individual column, or are used without subscripts if applied to entire tables. For example, an average count of the word "learning" in documents cited from a learning example document *d*, a potentially useful type of feature in document classification, is denoted as:

$$ave_{cnt}[\sigma_{word='learning' \wedge from='d'}$$
$$(Citation \bowtie_{to=doc} WordCount)]$$

The duplicates in the column *cnt* are not eliminated, unless an explicit projection of that column is performed before aggregation takes place. When ambiguous, column names are resolved with relation names. We abbreviate relation names with their first letter, and in the cases of joins involving more than one instance of the same relation, the relation name is suffixed with a numeral. For example, the number of common documents that both documents d^1 and d^2 cite is:

$$count[\sigma_{C1.from='d^1' \wedge C2.from='d^2'}(C1 \bowtie_{C1.to=C2.to} C2)].$$

This feature is an example of a feature useful in link prediction. It asks a question about a target pair of documents $< d^1, d^2 >$. When learning n-ary targets, we superscript d with a corresponding attribute index. Queries may be about just one of the documents in a target pair. For example:

$$count[\sigma_{to='d^2'}(C)]$$

is the number of times document d^2 is cited. Larger values of this feature increase the prior probability of d^1 citing d^2, regardless of what d^1 is.

2.1 Relational Feature Generation

We generate features by searching the space of relational database queries. The main principle of our search formulation is based on the concept of *refinement graphs* [Shapiro, 1983] which are widely used to search the space of first-order logic clauses. The search of refinement graphs starts with most general clauses and progresses by refining them into more specialized ones. Refinement graphs are directed acyclic graphs specifying the search space of the first-order logic queries. The space is constrained by specifying legal clauses (e.g. disallowing recursion and negation), and then structured by partial ordering of clauses, using a syntactic notion of generality (θ-subsumption [Plotkin, 1969]). A search node is expanded, or refined, applying *a refinement operator* to produce its most general specializations. Inductive logic programming systems using refinement graph search, usually apply two refinement operators: i) adding a predicate to the body of a clause, involving one or more variables already present, and possibly introducing one or more new variables, or ii) a single variable substitution, [Dzeroski and Lavrac, 2001]. In relational algebra these refinements correspond to adding an equijoin with a new relation instance, or performing a selection based on a condition of equality with a constant. In contrast to learning logic clauses, statistical learning is not limited to binary logic valued attributes.

Our first extension introduces aggregation or statistical operations into the search space. A query in relational algebra results in a table of all attribute values satisfying it, rather than a `true/false` value. Query results are aggregated to produce scalar numeric values to be used as features in statistical learning. Although there is no limit to the number of aggregation functions one may try, e.g. square root of the sum of column values, logarithm of their product etc., we expect a few of them being most useful, such as `count`, `ave`, `max`, `min`, `mode`, and `empty`. Aggregations can be applied to a whole table or to individual columns, as appropriate given type restrictions, e.g. `ave` cannot be applied to a column of a categorical type. Adding aggregation operators results in a much richer search space. Binary logic-based features are also included through the `empty` aggregation operation. The situations when aggregation operations are not defined, e.g. the average of an empty set, are resolved by introducing an interaction term with a 1/0 not-defined/defined feature.

The results of aggregation operations may be factored into further search. For example, we may want to ask how many co-authors the most cited author in a given conference c has (including him/herself). The following aggregation:

$$mostCitedAuth =$$
$$mode_{auth}[(\sigma_{vn='c'}((P \bowtie_{P.doc=from} C) \bowtie_{to=A.doc} A))]$$

is used in:

$$count[\pi_{A2.auth}(\sigma_{A1.auth='mostCitedAuth'}$$
$$(A1 \bowtie_{A1.doc=A2.doc} A2))].$$

Richer selection or join conditions, not necessarily conjuncts of equality conditions, can also be made part of the search space. The search space is potentially infinite, but not all subspaces will be equally useful. We propose the use of

[1] `http://citeseer.org/`

[2] Publication venues are extracted by matching information with the DBLP database: `http://dblp.uni-trier.de/`

[3] Domains, or types, used here are different from the basic SQL types. In implementation, these domains are specified in addition to the basic SQL types to guide the search process more efficiently.

sampling from subspaces of the same type performed at the time of node expansions to decide if more thorough exploration of that subspace is promising, or if the search should be more quickly refocused on other subspaces.

Our current implementation considers the search space covering queries with equijoins, equality selections and aggregation operations. Aggregates are considered for model inclusion at each node, but are not factored into the further search. Figure 1 presents a fragment of the search space using relations `Author`, `Citation` and `PublishedIn` for the link prediction task.

Logistic regression [Hosmer and Lemeshow, 1989] is used for binary classification problems. Model parameters/regression coefficients are learned to maximize the likelihood function, i.e. the probability that the training data is generated by a model with these coefficients. More complex models will results in higher likelihood values, but at some point will likely overfit the data, resulting in poor generalization. A number of criteria aiming at striking the balance between optimizing the likelihood of training data and model complexity have been proposed. Among the more widely used is the Bayesian Information Criterion (BIC) [Schwartz, 1979], which works by penalizing the likelihood by a term that depends on model complexity. We use stepwise model selection to find a model which generalizes well by adding one predictor at a time as long as the BIC can still be improved.

3 Tasks and Data

Learning from relational data for link prediction differs in several important aspects from other learning settings. Relational learning, in general, requires a quite different paradigm from "flat" file learning. The assumption that the examples are independent is violated in the presence of relational structure; this can be addressed explicitly [Jensen and Neville, 2002; Hoff, 2003], or implicitly, as we do here, by generating more complex features which capture relational dependencies. When the right features are used, the observations are conditionally independent given the features, eliminating the independence violation.

In our link prediction setting, a learning example class label indicating the presence or absence of a link between two documents is information of the same type as the rest of the link structure which can solely be used for prediction. In some sense, it may be instructive to view this setting as *hybrid model and memory based* learning. We build a formal statistical model, but prediction of future data points requires database access, as each selected feature is a database query. Thus, an important aspect, more so than in attribute-value learning, is what information about new examples will be available at the time of prediction and how missing or changing background information would affect the results.

Consider the following two scenarios for prediction of links between objects in a domain:

- The identity of all objects is known. Only *some* of the link structure is known. The goal is to predict unobserved links, from existing link structure alone or also using information about other available object attributes.
- New objects arrive and we want to predict their links to other existing objects. What do we know about new objects? Perhaps, we know some of their links, and want to predict the other. Alternatively, we might not know any of the links, but know some other attributes of the new objects.

In the latter case, when none of the partial link structure of the new objects is known, and prediction is based solely on other attributes, e.g. only authorship and word content, feature generation would have to be controlled to not produce features based on immediate links, but use them when referring to the links in already existing background knowledge.

In this paper, we perform experiments for the first scenario. The data for our experiments was taken from CiteSeer [Lawrence *et al.*, 1999]. CiteSeer catalogs scientific publications available in full-text on the web in PostScript and PDF formats. It extracts and matches citations to produce a browsable citation graph. The data we used contains 271,343 documents and 1,092,200 citations.[4] Additional information includes authorship and publication relations.[5] We use the following schema:

Citation(*from*:`Document`, *to*:`Document`),
Author(*doc*:`Document`, *auth*:`Person`),
PublishedIn(*doc*:`Document`, *vn*:`Venue`).

The training and test sets are formed by sampling citations (or absent citations for negative examples) from the citation graph. We perform learning on five datasets. Four of the datasets include links among all documents containing a certain query phrase, and the fifth data set covers the entire collection. Note that the background knowledge in the first four datasets also includes all other links in the full collection; only training and test links are sampled from the subgraph induced by document subsets. Table 1 contains the summary of the datasets.

The detailed learning setting is as follows:

- Populate three relations `Citation`, `Author` and `PublishedIn` initially with *all* data.
- Create training and test sets of 5,000 examples each by i) randomly sampling 2,500 citations for training and 2,500 citations for testing from those in column `# Links` of the Table 1; and ii) creating negative examples by sampling from the same subgraph also 2,500/2,500 train/test of "empty" citations, i.e. pairs of documents not citing each other.
- *Remove* test set citations from the `Citation` relation; but not the other information about the documents involved in the test set citations. For example, other citations of those documents are not removed.

[4] This data is part of CiteSeer as of August 2002. Documents considered are only non-singleton documents out of the total of 387,703. Singletons are documents which both citation indegree and outdegree registered in CiteSeer are zero.

[5] The authorship information is known for 218,313 papers, and includes 58,342 authors. Publication venues are known for 60,646 documents. The set of venues consists of 1,560 conferences and journals.

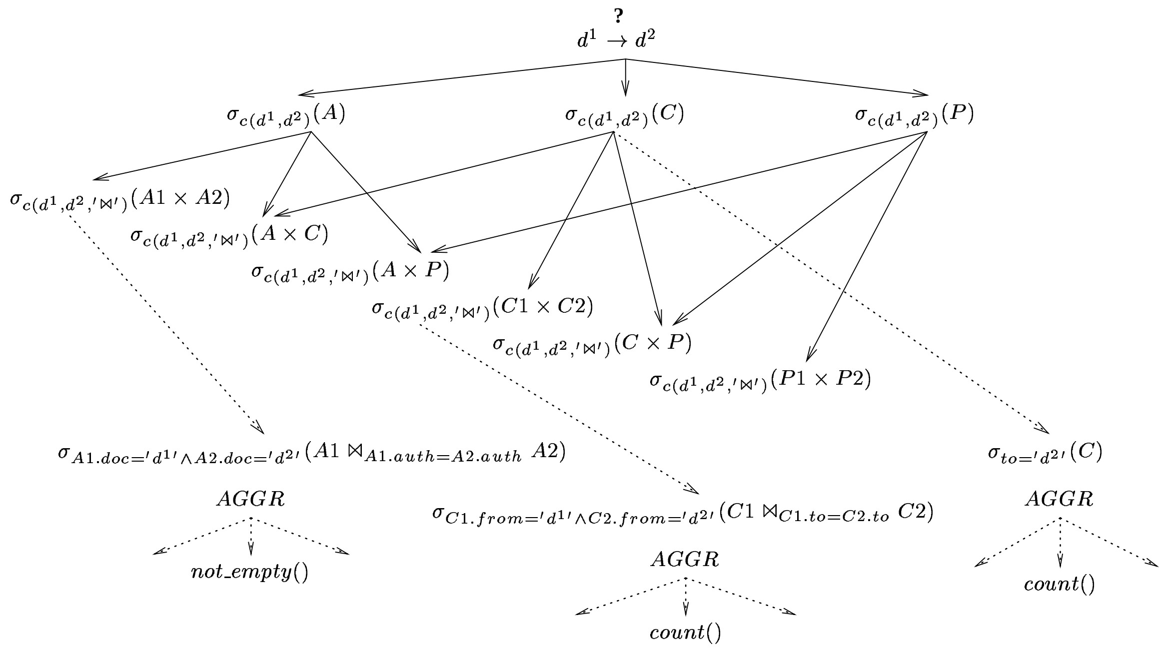

Figure 1: Fragment of the search space and examples. The select condition c is a boolean function specifying join conditions and equality conditions for referencing learning target-pair $< d^1, d^2 >$. Each node is a database query about d^1, d^2 or both (relation names are first letter abbreviations of `Citation`, `Author` and `PublishedIn` and the number if more that one instantiation of the same relation is involved. $AGGR$ denotes features resulting in aggregation operations at the corresponding search graph node).

Table 1: Dataset summaries.

Dataset	# Docs	# Links	Density (10^{-2}%)
"artificial intelligence"	11,144	16,654	1.3
"data mining"	3,424	6,790	5.8
"information retrieval"	5,156	8,858	3.3
"machine learning"	6,009	11,531	3.2
entire collection	271,343	1,092,200	0.1

- *Remove* training set citations from the `Citation` relation, so as not to include the actual answer in the background knowledge.
- Learning is performed i) using `Citation` relation only, or ii) using all three relations `Citation`, `Author` and `PublishedIn`.

The positive and negative classes in this task are extremely unbalanced. We ignore the lack of balance at the training phase; at the testing phase we perform additional *precision-recall* curve analysis for larger negative class priors. The next section reports experimental results.

4 Results

We start by presenting the results for the balanced class priors test scenario, and continue with the analysis of the unbalanced class settings. Two sets of models are learned for each dataset: i) using only `Citation` background knowledge, and ii) using all three relations `Citation`, `Author` and `PublishedIn`.

Table 2: Training and test set accuracies (%). 5,000 train/test examples; balanced priors.

Dataset	BK: Citation		BK: All	
	Train	Test	Train	Test
"artificial intelligence"	90.24	89.68	92.60	92.14
"data mining"	87.40	87.20	89.70	89.18
"information retrieval"	85.98	85.34	88.88	88.82
"machine learning"	89.40	89.14	91.42	91.14
entire collection	92.80	92.28	93.66	93.22

When only `Citation` background knowledge is used the average test set accuracy in five datasets is 88.73% and when all relations are used the average increases to 90.90%.[6] In both cases the search explored features involving joins of up to three relations. It is not unreasonable to expect that even better models can be built if we allow the search to progress further. Table 2 details the performance in each dataset. The largest accuracy of 93.22% is achieved for the entire CiteSeer dataset. Even though this is the largest and the most sparse dataset, this is not surprising because, since the features are not domain specific and rely on the surrounding citation structure, this dataset retains more useful "supporting link structure" after some of them are removed to serve as training and testing examples (Section 3).

In the experiments using only the `Citation` relation the average number of features selected is 32; 13 of the selected

[6] Using the predicted probability of 0.5 as the decision cut-off in logistic regression.

features are the same across all five datasets. When all three relations `Citation`, `Author` and `PublishedIn` are used the average number of selected features is 40, with 14 features common to all five datasets. In addition to more obvious features, such as d^1 is more likely to cite d^2 if d^2 is frequently cited, or if the same person co-authored both documents, or if d^1 and d^2 are co-cited, or cite the same papers[7], we learned some more interesting features. For example, a document is more likely to be cited if it is cited by frequently cited documents. Locally, this effectively learns the concept of an authoritative document [Page *et al.*, 1998; Kleinberg, 1999]. Or, the following feature, selected in all models:

$$count[\pi_{C2.to}(\sigma_{C1.to='d^{2}'}(C1 \bowtie_{C1.from=C2.from} C2))]$$

increases the probability of a citation if d^2 is co-cited with many documents. Since this feature is selected in addition to the simple citation count feature, it could mean that either d^2 appears more often in reviews, which tend to have longer lists of references, or it is cited from documents having smaller overlap among their references, which is more probable if they belong to different communities.

We compare the above results to the models trained on only binary features, i.e. when using only the `empty` aggregation function on the entire table. Such features are the logic-based features from the original formulation of refinement graph search; or, in other words, propositionalized inductive logic programming features with logistic regression feature selection [Popescul *et al.*, 2002]. The binary features resulted in models with lower out-of-sample accuracies in all datasets. On average the accuracy with only binary features is 2.52 percentage points lower in models using `Citation` relation, and 2.20 percentage points lower in models using all three relations. The decrease of accuracy is significant at the 99% confidence level in both cases according to the t-test.

The class priors are extremely unbalanced, due to the sparsity of the citation structure. The citation graph of the "artificial intelligence" dataset, for example is only 1.34×10^{-4} dense; that means that for one citation between two documents there are more than 7,000 non-existing citations, thus there are more than 7,000 times as many negative examples as there are positive. We perform the precision-recall curve analysis of our models trained with balanced class priors for testing situations with increased negative class proportions.

We vary the ratio k of the number of negative to the number positive examples used at testing. The ratio of one corresponds to the initial balance. We use for illustration the "artificial intelligence" dataset and the model trained using all three relations. New larger sets of negative examples are sampled with replacement from all "non-existing" links between documents in this dataset. Figure 2 presents precision-recall curves for k = 1, 10 and 100. As k increases the precision falls for the same levels of recall. Reducing the negative class prior should be performed when possible by filtering out obviously negative examples, for example by using a text based similarity, or other measures appropriate for a given task.

[7]These two features correspond to the concepts of co-citation and bibliographic coupling used in bibliometrics.

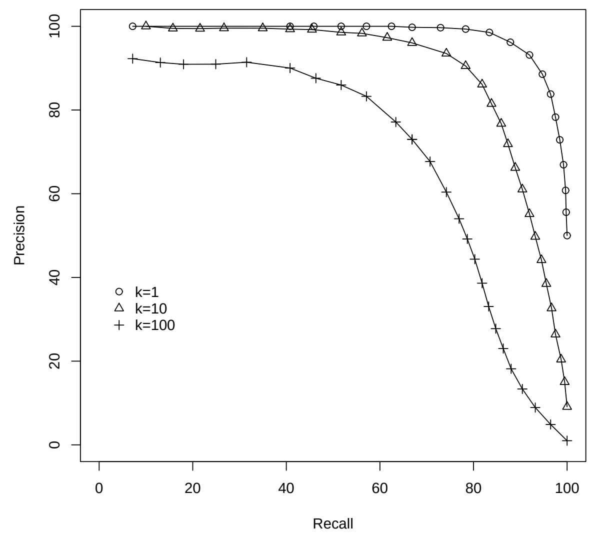

Figure 2: Precision-recall curves for the "artificial intelligence" dataset with different class priors. k is the ratio of the number of negative to the number of positive examples used at testing.

In application to citation recommendation, when only a few citations need to be recommended, we should care only about the high precision mode performance. In the case of the precision recall curve for k = 100, for example, 10% of citations can be recalled for recommendation with 91% precision. This is an overall measure of performance—some users can receive more than enough recommendations, and others none. When we want to recommend a fixed number of citations to *every* user, the CROC performance metric should be used [Schein *et al.*, 2002].

5 Related Work

Integrating or extending existing models or techniques to relational data has been addressed by researchers in several fields, including inductive logic programming, belief nets and link analysis.

A number of approaches extending one-table learners to multi-table domains have been proposed in the inductive logic programming (ILP) community. Generally, these approaches extend learners most suitable to purely binary attributes. Tilde [Blockeel and Raedt, 1998] and WARMR [Dehaspe and Toivonen, 1999], for example, extend decision trees and association rules, respectively. Another ILP approach is *propositionalization*. It uses bodies of first-order rules learned by an ILP technique as binary features in attribute-value learners. Kramer *et al.* [2001] review this methodology. Propositionalization with linear regression modeling, for example, was used by [Srinivasan and King, 1999] to build predictive models in a chemical domain. Decoupling the process of generating features by propositionalization and modeling using these features, however, retains the inductive bias of the technique used to construct features. As one solution, the generation of binary first-order features by an ILP-style search and feature selection with native criteria of a modeling technique are coupled into a single loop [Blockeel and Dehaspe, 2000; Popescul *et al.*, 2002], e.g. while modeling with a stepwise logistic regression. *Stochastic Logic Programs* [Muggleton, 1995] model uncertainly from within the ILP framework by providing logic theories with a probability distribution.

A number of probabilistic network relational models have also been proposed. *Probabilistic Relational Models* (PRMs) [Getoor *et al.*, 2001] are a relational version of Bayesian networks. PRMs are generative models of joint probability distribution capturing probabilistic influences between entities and their attributes in a relational domain. PRMs present a very powerful formalism. Being a joint probability model of the entire domain, PRMs can provide answers to a large number of possible questions about the domain, including class labels, latent groupings, changing beliefs given new observations. An important limitation, however, of generative modeling is that in reality there is rarely enough data to reliably estimate the entire model. One can achieve superior performance when focusing only on a particular question, e.g. class label prediction, and training models discriminatively to answer that question. A formulation similar to PRMs, but semantically different, called a *Statistical Relational Model*—a statistical model of a particular database instantiation—was proposed for optimizing the answering of relational database queries [Getoor *et al.*, 2002]. Taskar et al. [2002] propose a framework called *Relational Markov Networks* (RMNs)—a relational extension of Markov networks, trained discriminatively following the approach of Lafferty *et al.* [2001]. Here, the structure of a learning domain, determining which relational interactions are explored, is prespecified by a template expressed in a relational query language.

A technique called *Statistical Predicate Invention* [Craven and Slattery, 2001] combines statistical and relational learning by using classifications produced by Naive Bayes as predicates in FOIL [Quinlan and Cameron-Jones, 1995]. Statistical Predicate Invention preserves FOIL as the central modeling component and calls statistical modeling from within the inner structure navigation loop to supply new predicates. Neville and Jensen [2000] propose an iterative technique based on a Bayesian classifier that uses high confidence inferences to improve class inferences for related objects at later iterations. Cohn and Hofmann [2001] propose a joint probabilistic model of document content and connectivity, and apply it to classification tasks, including link prediction. A relational formulation of Markov chains for sequence modeling in web navigation is proposed in [Anderson *et al.*, 2002].

Link analysis plays an important role in the hypertext domains, a notable example being Google, which uses the link structure of the Web by employing a link based concept of page authority in ranking search results [Page *et al.*, 1998]. In addition to knowing the authoritative documents, it is often useful to know the web pages which point to authorities on a topic, the so called called "hub" pages [Kleinberg, 1999], which correspond to the concept of review papers in the scientific literature domain.

6 Discussion and Future Work

We presented the application of statistical relational learning to link prediction in the domain of scientific literature citations. The link prediction task is inherently relational. The noise in the available data sources suggests the use of statistical modeling. *Standard* statistical models, however, usually assume one table domain representation, which is inadequate for this task. Our approach overcomes this limitation. Statistical modeling and feature selection are integrated into a search over the space of database queries generating feature candidates involving complex interactions between objects in a given relational database. This avoids manual feature "packaging" into one table, a process that can be expensive and difficult.

Our method extends beyond ILP because statistics allows generation of richer features, better control of search of the feature space, and more accurate modeling in the presence of noise. On the other hand, our method differs from relational probabilistic network models, such as PRMs and RMNs, because these network models while being able to handle uncertainly, do not attempt to learn and model new complex relationships.

In addition to prediction, learned models can be used for explanatory purposes. Selected features provide insights into the nature of citations. Other linked environments, such as the Web, social networks or biological interactions, we believe, can be explored with the methodology presented in this paper.

We plan to use intelligent search heuristics to speed up the discovery of subspaces with more useful features. Since the potential search space is infinite, intelligent search is necessary to focus the process into more promising subspaces. As one approach, we propose using statistical estimates of "promise" computed by sampling from the subspaces of the same type to decide if those subspaces should be explored more thoroughly.

Using clustering or latent class modeling in statistical relational learning should also prove highly beneficial. Clusters can generate rich relational structure [Foster and Ungar, 2002]. For example, a document belongs to one or more topics. Each of these topics, in turn, has automatically generated properties such as "most frequently cited paper on this topic". Thus a feature such as `most-cited-doc(main-topic(doc-231))` could be learned, as could features involving sets of most cited documents. This has the potential to produce extremely rich and powerful models, helping to overcome problems of data sparsity.

Acknowledgments

The authors would like to thank Steve Lawrence and David Pennock for useful discussions. Small portions of the text also appear in [Popescul *et al.*, 2003].

References

[Anderson *et al.*, 2002] Corin Anderson, Pedro Domingos, and Dan Weld. Relational Markov models and their application to adaptive web navigation. In *Proceedings of the Eighth International Conference on Knowledge Discovery* *and Data Mining (KDD-2002)*, pages 143–152, Edmonton, Canada, 2002. ACM Press.

[Blockeel and Dehaspe, 2000] Hendrik Blockeel and Luc Dehaspe. Cumulativity as inductive bias. In P. Brazdil and A. Jorge, editors, *Workshops: Data Mining, Decision Support, Meta-Learning and ILP at PKDD-2000*, pages 61–70, 2000.

[Blockeel and Raedt, 1998] Hendrik Blockeel and Luc De Raedt. Top-down induction of logical decision trees. *Artificial Intelligence*, 101(1-2):285–297, 1998.

[Cohn and Hofmann, 2001] David Cohn and Thomas Hofmann. The missing link - A probabilistic model of document content and hypertext connectivity. In *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2001.

[Craven and Slattery, 2001] M. Craven and S. Slattery. Relational learning with statistical predicate invention: Better models for hypertext. *Machine Learning*, 43(1/2):97–119, 2001.

[Dehaspe and Toivonen, 1999] L. Dehaspe and H. Toivonen. Discovery of frequent datalog patterns. *Data Mining and Knowledge Discovery*, 3(1):7–36, 1999.

[Dzeroski and Lavrac, 2001] Saso Dzeroski and Nada Lavrac, editors. *Relational Data Mining*. Springer-Varlag, 2001. ISBN 3540422897.

[Foster and Ungar, 2002] Dean Foster and Lyle Ungar. A proposal for learning by ontological leaps. In *Proceedings of Snowbird Learning Conference*, Snowbird, Utah, 2002.

[Getoor *et al.*, 2001] L. Getoor, N. Friedman, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In Saso Dzeroski and Nada Lavrac, editors, *Relational Data Mining*. Springer-Verlag, 2001.

[Getoor *et al.*, 2002] Lise Getoor, Daphne Koller, and Benjamin Taskar. Statistical models for relational data. In *Proceedings of the Workshop on Multi-Relational Data Mining at KDD-2002*, pages 36–55, Edmonton, Canada, 2002.

[Hoff, 2003] P.D. Hoff. Random effects models for network data. In *Proceedings of the National Academy of Sciences: Symposium on Social Network Analysis for National Security*, 2003.

[Hosmer and Lemeshow, 1989] D.W. Hosmer and S. Lemeshow. *Applied logistic regression*. Wiley, New York, 1989.

[Jensen and Neville, 2002] D. Jensen and J. Neville. Linkage and autocorrelation cause feature selection bias in relational learning. In *Proceedings of the 19th International Conference on Machine Learning (ICML 2002)*, pages 259–266. Morgan Kaufmann, 2002.

[Kleinberg, 1999] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[Kramer *et al.*, 2001] S. Kramer, N. Lavrac, and P. Flach. Propositionalization approaches to relational data mining. In Saso Dzeroski and Nada Lavrac, editors, *Relational Data Mining*, pages 262–291. Springer-Verlag, 2001.

[Lafferty *et al.*, 2001] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML01*, 2001.

[Lawrence *et al.*, 1999] Steve Lawrence, C. Lee Giles, and Kurt Bollacker. Digital libraries and autonomous citation indexing. *IEEE Computer*, 32(6):67–71, 1999.

[Muggleton, 1995] S. Muggleton. Stochastic logic programs. In L. De Raedt, editor, *Proceedings of the 5th International Workshop on Inductive Logic Programming*. Department of Computer Science, Katholieke Universiteit Leuven, 1995.

[Neville and Jensen, 2000] J. Neville and D. Jensen. Iterative classification in relational data. In *Proceedings of AAAI-2000 Workshop on Learning Statistical Models from Relational Data*, pages 13–20. AAAI Press, 2000.

[Page *et al.*, 1998] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the Web. Technical report, Computer Science Department, Stanford University, 1998.

[Plotkin, 1969] G. Plotkin. A note on inductive generalization. In B. Meltzer and D. Michie, editors, *Machine Intelligence 5*, pages 153–163. Edinburgh University Press, 1969.

[Popescul *et al.*, 2002] Alexandrin Popescul, Lyle H. Ungar, Steve Lawrence, and David M. Pennock. Towards structural logistic regression: Combining relational and statistical learning. In *Proceedings of the Workshop on Multi-Relational Data Mining at KDD-2002*, pages 130–141, Edmonton, Canada, 2002.

[Popescul *et al.*, 2003] Alexandrin Popescul, Lyle H. Ungar, Steve Lawrence, and David M. Pennock. Statistical relational learning for document mining. Computer and Information Sciences, University of Pennsylvania, 2003. *http //www.cis.upenn.edu/~popescul/publications.html*.

[Quinlan and Cameron-Jones, 1995] J.R. Quinlan and R.M. Cameron-Jones. Induction of logic programs: FOIL and related systems. *New Generation Computing*, 13:287–312, 1995.

[Schein *et al.*, 2002] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25'th International ACM Conference on Research and Development in Information Retreival (SIGIR 2002)*, 2002.

[Schwartz, 1979] Gideon Schwartz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1979.

[Shapiro, 1983] E. Shapiro. *Algorithmic Program Debugging*. MIT Press, 1983.

[Srinivasan and King, 1999] A. Srinivasan and R. King. Feature construction with inductive logic programming: A study of quantitative predictions of biological activity aided by structural attributes. *Data Mining and Knowledge Discovery*, 3(1):37–57, 1999.

[Taskar *et al.*, 2002] Ben Taskar, Pieter Abbeel, and Daphne Koller. Discriminative probabilistic models for relational data. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI-2002)*, Edmonton, Canada, 2002.